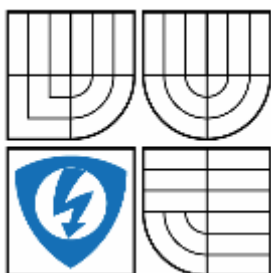




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

VIZUALIZACE KOMUNIKAČNÍCH TRAS V ENERGETICE

VISUALIZATION OF COMMUNICATION ROUTES IN THE ENERGETICS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Vít Pečeňa

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lešek Franek

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Vít Pečeňa

Ročník: 3

ID: 115253

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Vizualizace komunikačních tras v energetice

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit SW, který bude vizualizovat informace související s komunikací po energetickém vedení (PLC). Zobrazení topologie a úrovně signálu mezi jednotlivými zařízeními v mapách. Vizualizace změn jednotlivých parametrů v čase. Možnost filtrování a vyhledávání konkrétních dat z databáze.

DOPORUČENÁ LITERATURA:

BLANCHETTE, Jasmin a Mark SUMMERFIELD. C GUI programming with Qt 4. Upper Saddle River, NJ: Pearson Hall in association with Trolltech Press, c2006, xvii, 537 p. ISBN 978-013-1872-493.

JANAKA, Ekanayake. Smart Grid Technology and Applications. 2nd ed. Hoboken: John Wiley, 2012. ISBN 11-199-6868-2.

FRANEK, Lešek Data koncentrátor pro chytré sítě: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2012.

114 s. Vedoucí práce byl Ing. Pavel Kučera, Ph.D.

Termín zadání: 11. 2. 2013

Termín odevzdání: 27. 5. 2013

Vedoucí práce: Ing. Lešek Franek

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce se zabývá návrhem aplikace pro zobrazení pozic elektroměrů komunikující po elektrickém vedení (PLC). Aplikace by měla najít hlavní uplatnění v oblasti projektu Smart Grids (inteligentní sítě). Dále se v práci řeší výběr knihoven potřebný k tvorbě aplikace a její samotný návrh.

Klíčová slova

Knihovna Qt, C++, SQLite, QMapControl, Mapy, Geolokace, Inteligentní sítě, Přenos dat po elektrické síti, Vizualizace energetických sítí, Elektroměr

Abstract

This thesis describes the design of an application to view the positions electrometers communicating over Power Line (PLC). Applications should find the main use in the field of project Smart Grids. Furthermore, the work resolves selection of libraries required for creation of applications and the proposal itself.

Keywords

Qt Framework, C++, SQLite, QMapControl, Maps, Geolocation, Smart Grids, Power line communication, Visualization of energy networks, Electricity meter

Bibliografická citace:

PEČEŇA, V. *Vizualizace komunikačních tras v energetice*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 40 s. Vedoucí bakalářské práce Ing. Lešek Franek.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Vizualizace komunikačních tras v energetice jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **27. května 2013**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Lešku Frankovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **27. května 2013**

.....
podpis autora

Obsah

1.	Úvod.....	11
2.	Základy vývoje	12
2.1	Možnost zobrazení dat	12
3.	SMART GRID.....	13
3.1	Smart Grids v ČR.....	15
3.2	Struktura Smart Grids	15
3.2.1	Vodoměry, plynoměry, aj.	15
3.2.2	Zobrazovací zařízení	16
3.2.3	Elektroměry	16
3.2.4	Data-koncentrátory	16
3.2.5	Servery.....	17
4.	Podrobné zadání.....	18
4.1	Mapy	18
5.	Knihovna QT	20
5.1	Moduly	20
5.2	Mapový widget.....	20
5.2.1	QMapControl.....	20
5.2.2	Marble.....	21
5.2.3	MapView	21
5.2.4	Zvolené řešení	21
6.	QMapControl.....	22
6.1	Popis.....	22
6.2	Způsob načítání mapových podkladů.....	22
6.3	Podporované mapy.....	23
6.4	Práce s QMapControl	26
6.4.1	Postup vytvoření mapy	26
6.4.2	Vytvoření bodů na mapě	27
7.	Další využití technologie.....	28
7.1	SQLite	28
7.2	JavaScript Object Notation	28
7.3	Google API.....	29
7.3.1	Google Maps Geocoding API	29
8.	Program Energetické Sítě.....	30
8.1	Hlavní okno programu	30
8.1.1	Hlavní nabídka.....	31
8.2	Dialog „Najít na mapě“	31
9.	Vnitřní funkce programu	32
9.1	createActions()	32
9.2	createMenus() a createToolbars()	32

9.3	mapproviderSelected()	32
9.4	openLoadData().....	32
9.5	fillLocationQuery() a fillLocation()	34
9.6	wheelEvent() a keyPressEvent()	35
9.7	createPoints().....	35
10.	Závěr	37
11.	Literatura.....	38

A. Seznam obrázků

Obrázek 3-1: "Inteligentní měřidlo" – Smart Meter [2].....	13
Obrázek 3-2: Cílový stav Smart Grids [1]	14
Obrázek 3-3: Data-koncentrátor MT-54A [16]	16
Obrázek 3-4: Pyramidová struktura technologie [9].....	17
Obrázek 6-1: Aplikace s QMapControl na GPS zařízení [5].....	22
Obrázek 6-2: Mapová dlaždice ze serveru OpenStreetMaps	23
Obrázek 6-3: Česká republika v GoogleMaps	23
Obrázek 6-4: Česká republika v GoogleMaps (hybridní snímky).....	24
Obrázek 6-5: Česká republika v OpenStreetMap	24
Obrázek 6-6: Česká republika v YahooMaps	25
Obrázek 6-7: Česká republika v YahooMaps (satelitní snímky).....	25
Obrázek 6-8: Česká republika ve WMS mapách.....	26
Obrázek 8-1: Hlavní okno programu	30
Obrázek 8-2: Dialog Najít.....	31
Obrázek 9-1: Vývojový diagram funkce openLoadData()	33
Obrázek 9-2: Vývojový diagram funkce fillLocationQuery() a fillLocation().....	34
Obrázek 9-3: Vývojový diagram funkce createPoints()	36

B. Seznam tabulek

Tabulka 5-1: Srovnání mapových widgetů	21
--	----

1. ÚVOD

Cílem bakalářské práce je vytvoření aplikace s názvem „Energetické sítě“, která má za úkol vizualizovat informace související s komunikací po energetickém vedení (PLC). Zobrazení topologie a úrovně signálu mezi jednotlivými koncovými body v mapách.

Aplikace by měla najít hlavní uplatnění v oblasti projektu Smart Grids (inteligentní sítě), skládající se hlavně z elektroměrů Smart Meter, které budou v aplikaci zobrazeny.

Pro vytvoření aplikace je potřeba zvolit vhodné nástroje pro vývoj, nastudovat jejich použití, vlastnosti a práci s nimi. K volbě nástrojů přistupujeme s ohledem na požadavky, kterými jsou knihovna s GUI, obsáhlou dokumentací, funkčností na základních platformách a widget pro tvorbu map s podporou více poskytovatelů a rozsáhlou dokumentací.

Projekt je realizován ve spolupráci s firmami ModemTec, ČEZ a PRE.

Poskytnutá data firem s informacemi o aktuálním stavu elektroměrů budou zpracovávána databází. Vytvořená aplikace pak bude schopna s touto databází dále pracovat a zprostředkovávat uživateli prostřednictvím mapového rozhraní.

Uživatel tak bude moci na mapě sledovat umístění elektroměrů (resp. vodoměrů, plynoměru a jiných měřičů v síti), jejich stav a připojení k jeho nadřazeným data-koncentrátorům, které komunikují prostřednictvím PLC nebo bezdrátových sítí.

Získaná data pak mohou být dále využita například k detekci poruchy a hledání optimálních alternativ k zachování sítě v provozu, nebo ke zpracování krátkodobých i dlouhodobých statistik sloužících k dokonalejším budoucím optimalizacím systému jako celku. Vizualizace energetických toků pak zásadním způsobem přispěje k hledání snadnějších cest optimalizace celé infrastruktury.

2. ZÁKLADY VÝVOJE

Základní myšlenkou vývoje aplikace je přehledná správa energetických zařízení, vhodné zobrazení sesbíraných dat a jejich následné zpracování.

Aplikace má velký potenciál v oblasti regulace energetických toků, detekce poruch a přemostění sítě alternativní cestou nebo snížení ztrát z výroby nadbytečné energie.

2.1 Možnost zobrazení dat

Pro pokračování v práci bylo třeba zvolit optimální možnost vyhodnocování dat tak, aby se v nich technik zorientoval v krátkém časovém horizontu a zjistit stav zařízení. Pro velké množství dat, se kterými aplikace operuje, není vhodné vyhodnocovat výsledky tabulkově. V případě zobrazení výsledků v grafu by nastal problém identifikovat zařízení, ze kterého jsou data získávána. Vhodným způsobem reprezentace dat je mapa, kde lze zobrazit i interakce mezi elektroměry a data-koncentrátory. Pro dosažení cíle práce jsou vhodné webové a desktopové aplikace – použita byla varianta desktopového softwaru.

3. SMART GRID

Projekt Smart Grids („inteligentní sítě“) je součástí inovačního programu FUTUR/E/MOTION (www.futuremotion.cz), kterým společnost ČEZ reaguje na současné trendy Evropské Unie. [1]

Jako projekt Smart Grids se označuje samostatně pracující elektrorozvodná síť, která dokáže sama reagovat na změnu zátěže a přizpůsobovat výrobu energie. Tato technologie je schopna do celého systému přidat i měření dodávek plynu, vody a tepla. Její výhodou je průběh obousměrné komunikace mezi spotřebitelem a dodavatelem energie. Pomocí speciálních „inteligentních měřidel“ – Smart Meter, se sleduje spotřeba energií v budovách a provádí odečet této energie. [12]



Obrázek 3-1: "Inteligentní měřidlo" – Smart Meter [2]

Projekt Smart Grids je velkou výhodou pro spotřebitele, kteří získají přehled o využití elektrických energií v jejich domácnostech. Pro spotřebitele jsou tato data dostupná prakticky ihned a díky tomuto přehledu si spotřebitel bude moci optimalizovat zaplacené služby v jeho vlastní prospěch. Aktuálním odečtem spotřebovaných energií nebude nutné platit měsíční zálohy. [1]



Obrázek 3-2: Cílový stav Smart Grids [1]

1. Velkokapacitní elektrárny
2. Alternativní zdroje energie (větrné farmy, solární panely – elektřina z obnovitelných zdrojů slouží k vyrovnání poptávky a nabídky v síti)
3. Kogenerační jednotka v místě spotřeby energie
4. Elektromobil (infrastruktura veřejných dobíjecích stanic)
5. Automatizované kontrolní centrum (ovládá síť na základě informací ze sítě získaných v reálném čase)
6. Smart Home (automatické zapojování a odpojování spotřebičů k distribuční soustavě, možnost odložit spotřebu elektrické energie mimo špičku odběru elektřiny)
7. Smart Meter (instalace měřidel umožní obousměrný přenos informací a začlenění zákaznických elektrických zařízení přímo do sítě)
8. Elektromobil (může sloužit i jako akumulátor energie – nabíjení umožňuje vyrovnávání sítě tak, že se z připojených baterií uskladněná energie zpětně odčerpá)

9. Skladování energie (elektřina vyrobená v době nižší spotřeby může být uložena v bateriích a spotřebována ve špičce)
10. Dálkové ovladače a senzory (detekují kolísání a poruchy v distribuční soustavě a mohou automaticky izolovat postižené části)
11. Izolovaná část distribuční sítě (minimalizace možných poruch a výpadků díky možnosti flexibilně přesměrovat tok elektřiny, a izolovat tak postižené místo)

Pro spotřebitele tím vzniká hned několik nezanedbatelných výhod. Je schopen sledovat spotřebu v reálném čase a v případě neobvyklých odběrů je optimalizovat úpravami spotřebičů (obecně zařízení), nebo koupí nových s nižší spotřebou energie. Z historie spotřeby může spotřebitel zpracovat vlastní analýzu eventuálně podniknout opatření k optimalizaci vlastní domácnosti. Vzhledem k měření v reálném čase bude možné upravit platební tarif. Jednou z možností je vytvoření kreditu u dodavatele energií.

Dodavatelé a správci distribučních sítí energií budou schopni ovládat měřicí jednotky pomocí vzdáleného přístupu. Nebude nutné, aby zaměstnanci prováděli odečet stavu osobně, a tím lze ušetřit nejen čas, ale i peníze. Díky mapování jednotlivých jednotek bude možné kontrolovat odběry z jednotlivých nadefinovaných lokalit a na základě těchto dat také provádět dlouhodobou statistiku. Infrastruktura je rovněž nakloněná adaptaci alternativním způsobům získávání a distribuci energií.

3.1 Smart Grids v ČR

V roce 2010 spustila společnost ČEZ první testování sítí Smart Grids v regionu Vrchlabí, v okolí Hradce Králové a Pardubic. Na přibližně 40 000 nainstalovaných inteligentních elektroměrech má probíhat testování až do roku 2015. [1]

Vrchlabí byla ideální volba z důvodu existence lokálních zapojitelných obnovitelných zdrojů energie i díky prostoru pro zapojení několika elektroměrů s kombinovanou výrobou elektřiny a tepla.

3.2 Struktura Smart Grids

3.2.1 Vodoměry, plynoměry, aj.

Do této kategorie spadají nejen zařízení měřící různé druhy energie kromě elektrické, ale i jiná zařízení získávající data (průmyslové teploměry, hladinoměry ...)

Zařízení s elektroměry komunikují pomocí ZigBee, Wireless M-BUSu nebo metalického M-BUSu. K přenosu dat dochází směrem ze zařízení do elektroměru, některá zařízení dokonce mohou přijímat data pro jejich konfiguraci. [9]

3.2.2 Zobrazovací zařízení

Prostřednictvím rozhraní zobrazující aktuální, ale i historická data odběrů má spotřebitel možnost zasáhnout do platebních tarifů nebo i zadávat příkazy, jako může být přerušení odběru při cestě na dovolenou apod.

3.2.3 Elektroměry

Elektroměry měří kromě spotřeby elektrické energie i další užitečná data – maximální hodnoty výkonu, napětí v příslušném místě nebo různé typy událostí. Zařízení je schopno vykonávat úlohy jako jsou změny tarifu, spínání relé v závislosti na tarifu, omezení nebo i úplné odpojení odběru ze sítě.

S data-koncentrátory komunikují nejčastěji prostřednictvím PLC sítě. V místech, kde selhává tato síť, se data z elektroměrů odesílají přímo na server prostřednictvím GPRS. Další možností připojení k data-koncentrátoru je rádiový přenos anebo prostřednictvím součtového elektroměru, který využívá spojení přes M-BUS, RS-485, Ethernet a jiná rozhraní. [9]

3.2.4 Data-koncentrátory

Data-koncentrátory se nacházejí přímo v trafostanicích a slouží jako rozhraní k přenosu dat po elektrické nebo rádiové síti či přes TCP/IP. Jeden data-koncentrátor obsluhuje přibližně 100 zařízení (v některých případech velkých sídlišť až 1000 zařízení). Komunikace se serverem probíhá bezdrátovým přenosem nebo prostřednictvím místní „drátové“ sítě, v některých místech kde nejsou tato spojení možná, se využívá GPRS. [9]



Obrázek 3-3: Data-koncentrátor MT-54A [16]

3.2.5 Servery

Servery sesbírají veškerá data zasláná z data-koncentrátoru. Tato data zpracovávají a poskytují je operátorům sítě, kteří zasílají elektroměrům příkazy ke změně jejich stavu, aby předešli možnému kolapsu sítě, nebo při přebytku energie upravili tarif spotřebiteli. Některá data jsou zaslána zpět na zobrazovací stanice klientů k dalším operacím. [9]



Obrázek 3-4: Pyramidová struktura technologie [9]

4. PODROBNÉ ZADÁNÍ

Výsledná aplikace má sloužit k zobrazení, diagnostice kvality komunikačních tras a možnost pozměnit nastavení tras k dosažení kvalitnějších výsledků. Aplikace po spuštění načte mapové podklady ze serveru a umístění elektroměrů s data-koncentrátory z databáze a vykreslí je na daná místa na mapě. V dalším kroku bude zjištěna kvalita spojení mezi zařízeními a také vykreslena do mapy.

Na každém data-koncentrátoru bude možnost nastavení, která data budou odesílána na server, která budou pouze na vyžádání a která nebudou uchovávána vůbec.

Příklad: Data-koncentrátor odesílá kvalitu komunikace každého měřiče za celý den, shromažďuje hodinové hodnoty v pracovní dny a dny klidu u sebe, dodatečně je možné zapnout odesílání průměru za určité období např.: hodina nebo odesílání všech hodnot. Pokud aplikace zjistí, že potřebná data, nejsou v databázi, nabídne uživateli možnost získat údaje z data-koncentrátoru. V případě, že potřebné naměřené údaje neobsahuje ani data-koncentrátor, aplikace událost oznámí uživateli.

4.1 Mapy

Pomocí dostupných map a GPS souřadnic všech zařízení nebo poštovní adresy, budou zobrazeny objekty v mapě. Objekty budou vizualizovány podle aktuální topologie se svými nadřazenými prvky.

Zobrazované údaje budou reprezentovány zejména pomocí barvy jednotlivých objektů.

Mapové podklady:

- standartní silniční mapy (více typů pro porovnání)
- satelitní snímky

Zobrazované údaje týkající se zařízení:

- kvalita spojení s jednotlivými zařízeními, signalizována barvou zařízení
- datum posledních vyčtených dat
- výskyt události – četnost
- výskyt události – časově

Zobrazované údaje týkající se cest:

- kvalita spojení na příslušné cestě
- veškerá jiná možná spojení, včetně jejich kvality

Filtrování podle:

- projektu
- typu zařízení
- oblasti
- skupiny
- zobrazované hodnoty
- skrytí cest

Vyhledávání podle:

- čísla zařízení
- adresy
- GPS souřadnic

Zobrazení údajů po kliknutí na zařízení:

- základní údaje o zařízení
- možnost zobrazit všechna možná spojení od zařízení
- nastavení routingu

Další možnosti:

- zobrazení vlastních mapových podkladů – obrázky s trasami skutečných energetických tras
- animace změny stavu v zadaném rozsahu data a času
- nastavování skupiny podle oblasti v mapě

5. KNIHOVNA QT

Program Energetické sítě je vytvořen pomocí knihovny Qt, která je napsána v programovacím jazyce C++. Qt se vyslovuje jako /'kju:t/, což zní v angličtině stejně jako slovo „cute“, které v překladu znamená „roztomilý“. Tato knihovna byla vytvořena roku 1999 norskou softwarovou společností Trolltech a v roce 2008 prodána firmě Nokia. [3]

V současnosti patří mezi nejpoblárnější multiplatformní knihovny s GUI pro vytváření programů. Její výhoda tkví v tom, že vytvořené programy jsou spustitelné na různých operačních systémech bez nutnosti přepisování kódu. Qt funguje na všech základních platformách – Windows, OS X a Linux.

Je to knihovna programovacího jazyka C++, ale existuje i pro jazyky jiné, jako např. C, C#, Ruby, Python, Pascal, Java ... Mezi jeho další vlastnosti patří i podpora lokalizace a SQL, práce se soubory a multimédií, zpracování XML.

Qt má velmi přehledně zpracovanou dokumentaci dostupnou na internetu. Součástí jsou dvě integrovaná vývojová prostředí. Qt Creator, který obsahuje editor zdrojového kódu a kompilátor. Qt Designer, v kterém se navrhuje vzhledy GUI.

V Qt jsou udělány známé projekty jako například Opera, Skype, Google Earth, desktopové prostředí operačního systému Linux KDE a další.

Mezi základní a velmi využívané nástroje Qt je mechanismus signál/slot. Tento mechanismus je používán pro komunikaci mezi objekty a je to hlavní výhoda tohoto prostředí oproti jiným vývojovým nástrojům. [7]

5.1 Moduly

Knihovna Qt obsahuje několik modulů, pro tvorbu určitých aplikací např.:

- QSql – zabývající se vytvářením databází
- QtXML – pro četbu a tvorbu XML souborů
- QtOpenGL – možnost využití 3D grafiky
- QtNetwork – přidá možnost využívat komunikaci s FTP a http

5.2 Mapový widget

Pro vytvoření mapy v Qt aplikaci bylo zapotřebí získat knihovnu, která bude tento mapový widget podporovat. Na výběr bylo z několika takových knihoven QMapControl, Marble a MapView.

5.2.1 QMapControl

Knihovna QMapControl vznikla jako diplomová práce německého studenta Kai Wintera. Má velmi dobrou dokumentaci a za pomoci vzorových příkladů se dá lehce pochopit. QMapControl podporuje různé poskytovatele mapových podkladů (podrobnější popis QMapControl viz. Kapitola 6.) [5]

5.2.2 Marble

Marble je knihovna z KDE API (desktopové prostředí pro Linux). Má velice rozsáhlou dokumentaci a pracuje s OpenStreetMaps. Pro tento projekt je ale až příliš složitá. [10]

5.2.3 MapView

MapView je základní widget Qt pro zobrazení mapy. Je to ale jen čistě mapa, bez žádných dalších funkcí (ty si musí uživatel napsat sám). [11]

5.2.4 Zvolené řešení

Vzhledem k počtu výhod byl QMapControl zvolen jako nejlepší volba mapového widgetu pro tuto aplikaci. Nepodporuje sice offline mapy, ale to kompenzuje větší podporou mapových poskytovatelů.

Tabulka 5-1: Srovnání mapových widgetů

	QMapControl	Marble	MapView
Dokumentace	✓	✓	✓
Vzorové příklady	✓	✗	✗
Podpora více mapových zdrojů	✓	✗	✗ ¹
Offline mapy	✗	✓	✗

¹ Uživatel si může dopsat sám funkci pro podporu jiného zdroje

6. QMAPCONTROL

6.1 Popis

QMapControl je Qt Widget, umožňující v aplikaci zobrazit mapové podklady. Do těchto map se následně mohou zakreslovat různé objekty (body, linky, obrázky i jiné Qt Widgety). Je možné vytvořit i více vrstev s různými mapovými podklady a kombinovat je (tvorba hybridních map).

QMapControl umí pracovat i s GPS přijímačem a vyhledávat na mapě jeho aktuální pozici.



Obrázek 6-1: Aplikace s QMapControl na GPS zařízení [5]

Program Energetické sítě je založen hlavně na QMapControl, s jeho pomocí se vykresluje mapa a i důležité objekty na ni (elektroměry i důležitá spojení mezi nimi).

6.2 Způsob načítání mapových podkladů

Celá mapa se skládá z čtvercových dlaždic, které aplikace získává ze serveru poskytovatele. Požadavky například na server OpenStreetMaps obsahují tři parametry: zoom, poloha x a poloha y. Tyto parametry, odeslané na server například ve tvaru „<http://tile.openstreetmap.org/11/1113/694.png>“, vrací programu dlaždici s mapou Pardubic (viz Obrázek 6-2). Ostatní servery fungují obdobně, jen požadavek posílají v jiném tvaru.



Obrázek 6-2: Mapová dlaždice ze serveru OpenStreetMaps

Jednotlivé dlaždice poté QMapControl seřadí vedle sebe a vykreslí požadovanou oblast v okně aplikace. V případě posunutí mapy se stáhnou další potřebné dlaždice a ty se připojí k těm předchozím. Při přiblížení (oddálení) se musí všechny dlaždice stáhnout znovu, ale již s novým parametrem zoomu.

6.3 Podporované mapy

QMapControl podporuje různé mapové poskytovatele, bohužel je již poněkud zastaralý a v aktuální době někteří poskytovatelé změnili způsob odesílání dat ze serveru a tak v současnosti nejsou v provozu.

GoogleMaps

- web - <http://maps.google.com/>
- celosvětově nejznámější a nejpoužívanější mapa
- zdarma pro nekomerční využití
- nevýhodou je omezení na 2500 žádostí za 24 hodin na IP adresu
- hybridní mapa je velmi kvalitní a podrobná



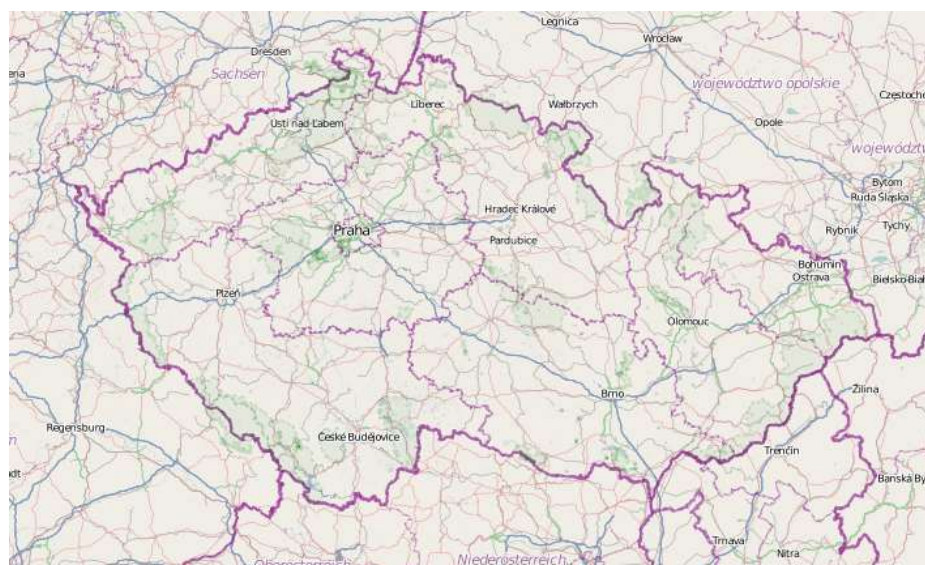
Obrázek 6-3: Česká republika v GoogleMaps



Obrázek 6-4: Česká republika v GoogleMaps (hybridní snímky)

OpenStreetMap

- web - <http://www.openstreetmap.org/>
- open-source projekt, jehož cílem je tvorba volně dostupných geografických dat a následně jejich vizualizace do podoby topografických map za využití záznamů z GPS přijímačů
- **výhody:**
 - neustále aktualizovaná „lidmi pro lidi“
 - otevřená licence CC-BY-SA
 - žádné denní limity



Obrázek 6-5: Česká republika v OpenStreetMap

YahooMaps

- web - <http://maps.yahoo.com/>
- využívá vektorová data od Navteq, Tele Atlas a z veřejných zdrojů
- **výhody:**
 - možnost použít klasické nebo satelitní snímky
- **nevýhody:**
 - omezení na 50 000 žádostí za 24 hodin na jednu IP adresu
 - pomalejší načítání
 - horší satelitní snímky



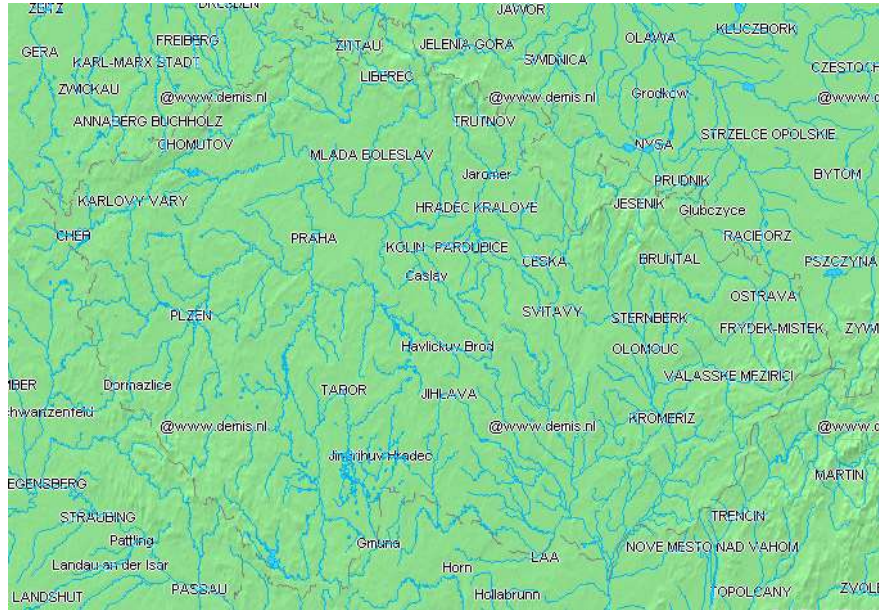
Obrázek 6-6: Česká republika v YahooMaps



Obrázek 6-7: Česká republika v YahooMaps (satelitní snímky)

WMS (Web Map Service)

- méně známý poskytovatel
- pro tento projekt nepoužitelná mapa, proto nebude použita
- **nevýhody:**
 - velmi jednoduché mapy
 - nezakreslena infrastruktura měst



Obrázek 6-8: Česká republika ve WMS mapách

6.4 Práce s QMapControl

6.4.1 Postup vytvoření mapy

Pro vytvoření mapy je potřeba v Qt projektu nejdříve vytvořit widget, v kterém se bude mapa generovat a to pomocí funkce `MapControl()`. Tato funkce se musí volat s parametrem `QSize(šířka, výška)`, který nastaví rozměry widgetu. Pro přidání měřítka se využívá příkaz `showScale(true)`. Nyní je potřeba tento widget zobrazit, a k tomu slouží příkaz `setCentralWidget()`.

Poté již stačí ve widgetu vytvořit vrstvu a tu propojit s map-adaptérem požadovaného poskytovatele. Základní kód pro zobrazení mapy poté bude vypadat takto:

```
// vytvoreni MapControl
mc = new MapControl(QSize(640,480));
// zobrazení měřítka
mc->showScale(true);
// zobrazení widgetu v aplikaci
setCentralWidget(mc);
// vytvoření OpenStreetMap adaptéru
mapadapter = new OSMMAdapter();
```

```
// vytvoření vrstvy obsahující mapový adaptér
layer = new MapLayer("OpenStreetMap-Layer", mapadapter);
// přidání vrstvy do widgetu
mc->addLayer(layer);
```

6.4.2 Vytvoření bodů na mapě

QMapControl umí do mapy zakreslit tzv. point (bod), tyto body mohou být buď skryté – *Point()*, kruhové – *CirclePoint()* anebo obrázkové – *ImagePoint()*.

Funkce mají tyto parametry:

- longitude – zeměpisná délka (povinný parametr)
- latitude – zeměpisná šířka (povinný parametr)
- poloměr a styl kruhu (jen *CirclePoint*)
- název bodu
- zarovnání
- cesta k obrázku (jen *ImagePoint*)

Příklad:

```
Point(16.300842, 49.124872, "skryty bod");
CirclePoint(16.500862, 49.143632, "kruhovy bod", Point::Middle);
ImagePoint(16.500862, 49.173632, ":/ele/red", "obrazkovy bod");
```

Mezi určenými body se přes funkci *LineString()* dají vykreslit spojnice.

7. DALŠÍ VYUŽITÉ TECHNOLOGIE

7.1 SQLite

SQLite je malá knihovna pro vytvoření lokální databáze. Celá tato databáze je poté uložena pouze v jednom souboru. Knihovna je vyvíjena v jazyce C programátorem D. Richardem Hippem.

Výhodou této knihovny je snadný přesun vytvořené databáze. Pro všechny operační systémy využívá stejný formát. U většiny běžných operací je rychlejší než jiné databáze, které jsou na principu klient/server (MySQL, PostgreSQL, aj.)

Menší problém při dlouhodobějším využívání této databáze ale spočívá ve fragmentaci dat.

Omezení SQLite databáze:

- velikost - až 2^{41} byte (2 TB)
- velikost jednoho řádku – max. 1 MB
- počet řádků pro jednu tabulku – až 4 biliony
- jména objektů – bez omezení
- jména funkcí – max. 255 znaků

Existují různí správci SQLite databáze např. aplikace SQLite Expert nebo rozšíření SQLite Manager pro internetový prohlížeč Mozilla Firefox. V těchto správcích si uživatel může nejen prohlížet data v databázi, ale má možnost i vytvářet nové databáze a tabulky v nich a poté je naplnit daty.

7.2 JavaScript Object Notation

JavaScript Object Notation, neboli JSON (česky JavaScriptový objektový zápis) je způsob zápisu dat (datový formát) nezávislý na počítačové platformě, určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. JSON je textový, na jazyce nezávislý formát jednoduše čitelný a zapisovatelný člověkem a snadno analyzovatelný či generovaný strojem. Vstupem je libovolná datová struktura a výstupem je vždy řetězec. Složitost hierarchie vstupní proměnné není teoreticky nijak omezena. JSON je pro výměnu dat nejvhodnějším jazykem, jelikož využívá konvence nejznámějších programovacích jazyků (C, C++, C#, Java, JavaScript, aj.) [13] [14]

Pro možnost využívání JSON v Qt aplikaci, je potřeba přidat knihovnu QJson, která jej převádí do QVariant (datové pole Qt).

Výhody:

- jednoduchý přenos dat
- čitelnost člověkem

Nevýhody:

- nemožnost definice znakové sady přenášeného obsahu
- nevhodné pro přenos binárních dat

Jako alternativa by se mohl použít jazyk XML, ten je ale datově náročnější.

7.3 Google API

Google API je sbírka JavaScriptových API vyvinutých společností Google, přidávající webovým stránkám či aplikacím možnost využívání aplikací od Google (mapy, email, kalendář a spoustu dalších). V této práci je použito Google Maps API, přesněji jeho část Google Maps Image API pro načítání mapy (způsob načítání popsán v kapitole 6.2) a Google Maps Geocoding API pro převod adres požadovaných míst na jejich zeměpisnou polohu.

7.3.1 Google Maps Geocoding API

Velkou nevýhodou je omezení na 2 500 geolokačních žádostí na den (verze Google Maps API pro business dovoluje až 100 000 žádostí na den).

Žádost na server obsahuje dva parametry – adresa nebo zeměpisná poloha a senzor. Senzor udává, jestli zařízení, z kterého byla žádost odeslána, obsahuje lokační senzor.

Návratové pole hodnot žádosti se vždy skládá ze dvou hlavních částí - stavová zpráva a výsledek.

Stavové zprávy

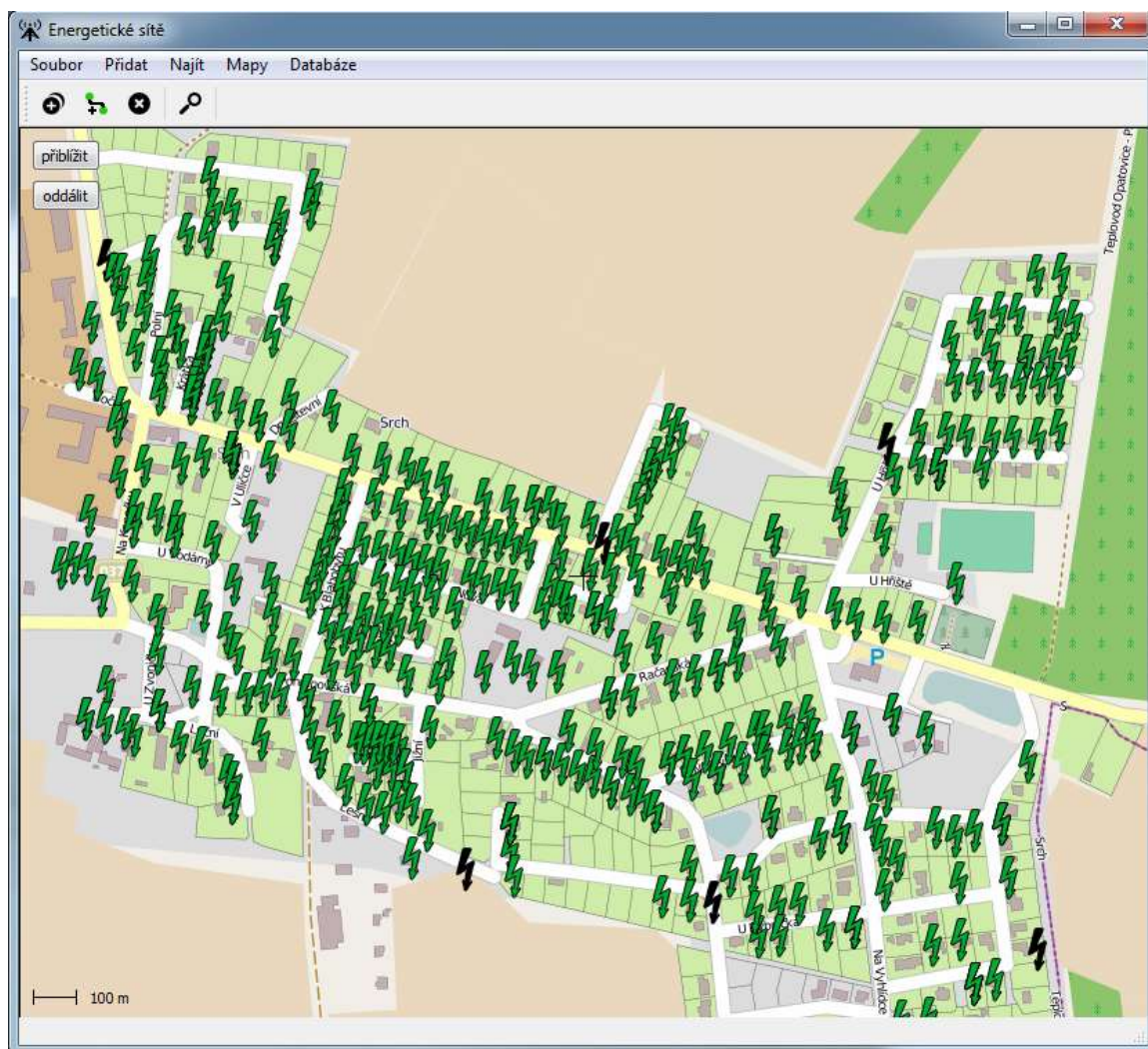
Využívají se pro zjištění úspěšnosti dotazu.

- OK – všechno proběhlo v pořádku, byl navrácen minimálně jeden výsledek
- ZERO_RESULT – dotaz byl podán správně, ale nevrátil žádný výsledek (například neexistující adresa)
- OVER_QUERY_LIMIT – přesáhnutí denního limitu žádostí
- REQUEST_DENIED – žádost zamítnuta, většinou z důvodu chybějícího parametru pro určení senzoru
- INVALID_REQUEST – chybí zadaná adresa nebo zeměpisná poloha
- UNKNOWN_ERROR – neznámá chyba

8. PROGRAM ENERGETICKÉ SÍŤE

8.1 Hlavní okno programu

Po spuštění programu se zobrazí hlavní okno s hlavní nabídkou, mapou, nástrojovou a stavovou lištou (viz *Obrázek 8-1*).



Obrázek 8-1: Hlavní okno programu

S mapou se pohybuje pomocí myši nebo klávesnice (pohyb do stran pomocí kláves W, S, A, D), přibližování a oddalování funguje přes rolování kolečka myši, stiskem tlačítek v levém horním rohu mapy anebo klávesami + a -. V levém dolním rohu se nachází měřítko.

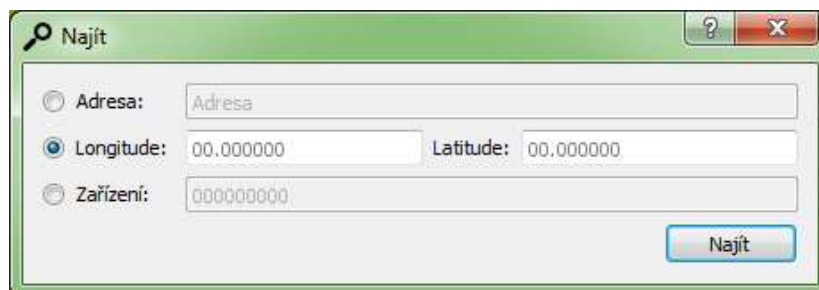
8.1.1 Hlavní nabídka

Hlavní nabídka je rozdělena do základních částí:

- soubor – obsahuje pouze tlačítko pro ukončení celého programu. Načtený soubor má v sobě uloženy vlastnosti elektroměrů ve formátu XML, které se pak zkopírují do datového pole a vykreslí na mapě
- přidat – operace s vykreslenými značkami, možnost vymazání z mapy a poté jejich znovunačtení (viz kapitola 9.5)
- najít – vyvolání dialogu pro nalezení určité adresy, zeměpisné polohy nebo zařízení na mapě (viz kapitola 8.2)
- mapy – výběr poskytovatele zobrazených map (OpenStreetMaps, GoogleMaps nebo YahooMaps)
- databáze – nabídka databáze obsahuje akci pro doplnění tabulky Resources (viz kapitola 9.5) a načtení souboru (nebo souborů) s daty o kvalitě spojení (viz kapitola 9.3)

8.2 Dialog „Najít na mapě“

Po otevření dialogu se zobrazí nabídka s dotazem, zda chce uživatel najít na mapě adresu, zeměpisnou polohu nebo přímo určité zařízení, poté potvrdí a náhled mapy, se plynule přesune do určené oblasti.



Obrázek 8-2: Dialog Najít

Jelikož QMapControl již má funkci pro vyhledávání, ale pouze pro zeměpisné souřadnice, musí se jiné zadané údaje převést. Varianty hledání:

- Vyhledávání adresy – uživatel zadá přesnou adresu (pro nejpřesnější výsledek se musí adresa zadat ve tvaru „ulice a číslo popisné, město, psč“), kterou chce vyhledat a tu poté program za pomoci Google API převede na zeměpisnou polohu
- Vyhledávání zeměpisné polohy – není nutnost dalších úprav a program přímo vyhledá zadanou polohu
- Vyhledávání podle id zařízení – program se pokusí vyhledat v databázi zadané identifikační číslo zařízení a pokud jej nalezne, načte z databáze zeměpisnou polohu

9. VNITŘNÍ FUNKCE PROGRAMU

9.1 createActions()

Funkce *createActions()* vytváří různé akce (vykreslení zařízení nebo cest, aj.), a zároveň nastavuje jejich parametry. Příklad kódu pro zavírání programu:

```
exitAction = new QAction(QString::fromLocal8Bit("Zavřít"), this);
exitAction->setShortcut(tr("Ctrl+Q"));
exitAction->setIcon(QIcon(tr(":/img/close")));
exitAction->setStatusTip(QString::fromLocal8Bit("Zavřít aplikaci"));
connect(exitAction, SIGNAL(triggered()), qApp, SLOT(closeAllWindows()));
```

První řádek vytvoří novou akci se jménem *exitAction* a přidá mu titulek „Zavřít“. Následujícím příkazem *setShortcut* se nastaví klávesová zkratka pro vyvolání akce, příkaz *setIcon* určuje ikonu a *setStatusTip* určí, co se zobrazí ve stavové liště. V posledním řádku je využito spojení signal / slot a ten programu oznamuje, že se při aktivaci akce *exitAction* má ukončit celý program.

9.2 createMenus() a createToolbars()

createMenus() vytváří hlavní nabídku a *createToolbars()* nástrojovou lištu do obou se poté přidávají akce vytvořené funkcí *createActions()*.

```
QMenu *menu;
menu = menuBar()->addMenu(tr("&Přidat"));
menu->addAction(createPointsAction);
menu->addAction(createRoutesAction);
menu->addSeparator();
menu->addAction(deletePointsAction);
```

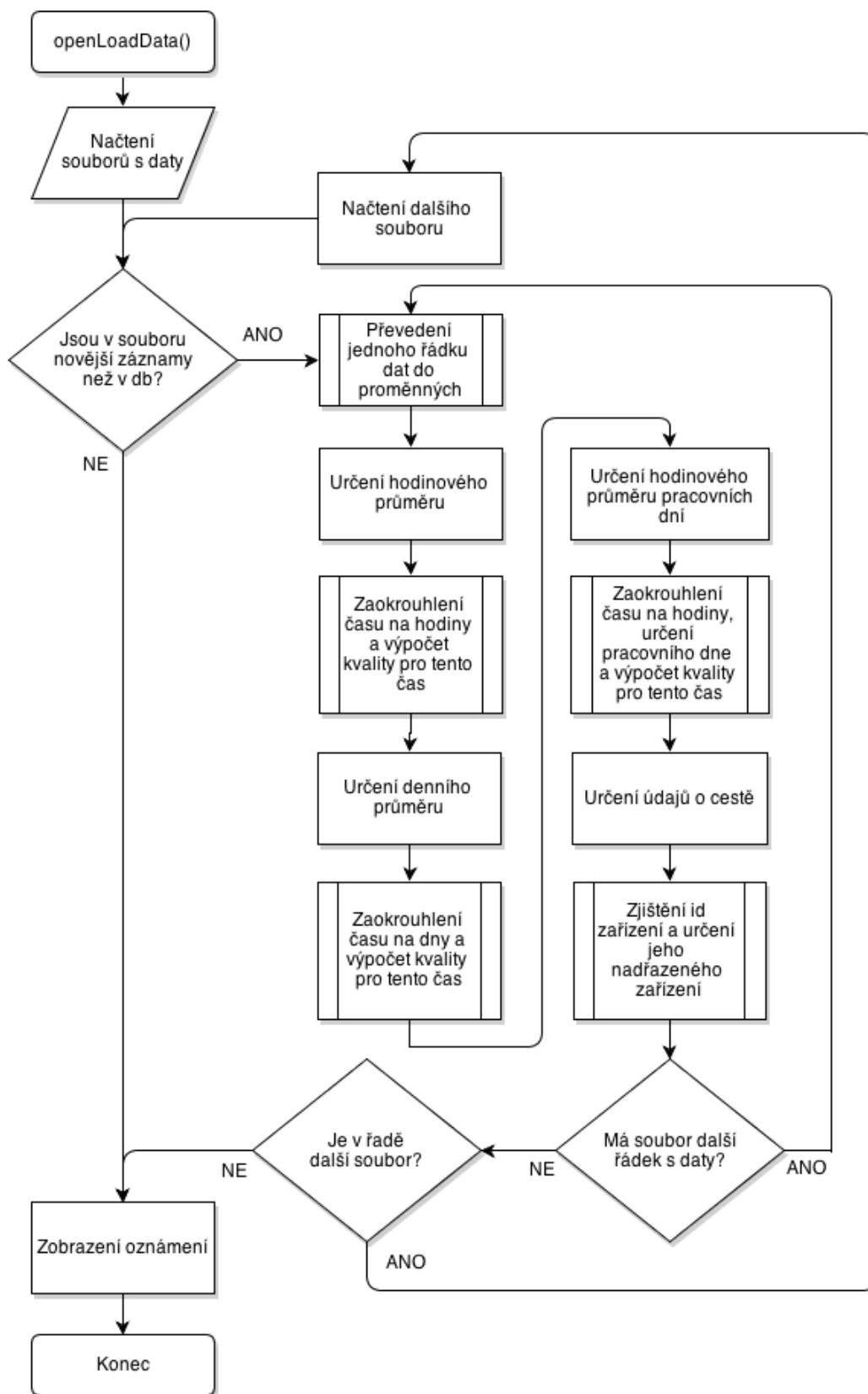
QMenu deklaruje proměnnou *menu* jako ukazatel na hlavní nabídku. Příkaz *menuBar()->addMenu()* vytvoří nabídku s názvem Přidat. Za pomoci *addAction()* se do nabídky přidávají položky. *addSeparator()* složí pro oddělování položek.

9.3 mapproviderSelected()

V horní části hlavního okna je nabídka Mapy, při výběru poskytovatele (viz kapitola 6.2) se odešle akce do funkce *mapproviderSelected()* kde se změní vrstva mapového adaptéru, překreslí mapu a vypíše informaci o změně do stavové lišty.

9.4 openLoadData()

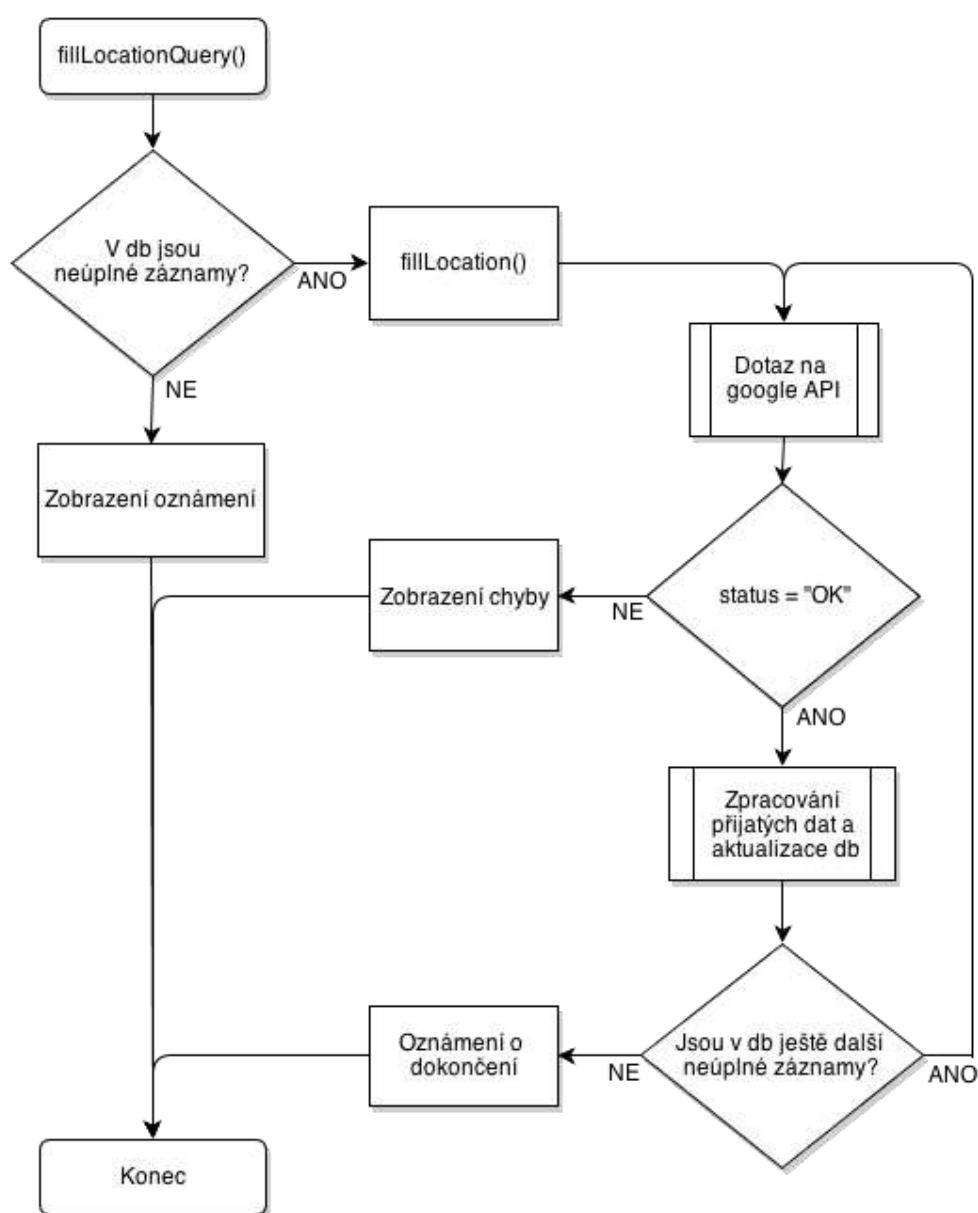
Při spuštění funkce *openLoadData()* se objeví okno pro výběr jednoho či více souborů, které se po řádcích postupně zpracují. Každý řádek se skládá z několika hodnot oddělených středníky. Tyto hodnoty se načtou do proměnných, s kterými pracují další podfunkce (diagram popisující funkci *openLoadData()* viz Obrázek 9-1).



Obrázek 9-1: Vývojový diagram funkce openLoadData()

9.5 fillLocationQuery() a fillLocation()

Funkce *fillLocationQuery()* slouží jen ke kontrole, zda jsou v databázi neúplné záznamy. Uživateli sdělí počet těchto záznamů a poté spustí funkci *fillLocation()*, která z databáze postupně načítá řádky s neúplnými záznamy. Mohou nastat dvě situace, chybí zeměpisná poloha nebo celá adresa. Funkce zašle dotaz na Google API se známými údaji, ten následně vrátí odpověď ve formátu JSON (viz kapitola 7.2), obsahující veškeré záznamy o dotazovaném místě a informaci o úspěšnosti dotazu (status). Pokud je vše v pořádku, záznam se v databázi aktualizuje již s kompletními informacemi, v jiném případě je uživateli oznámena chyba (diagram popisující tyto funkce viz Obrázek 9-2).



Obrázek 9-2: Vývojový diagram funkce *fillLocationQuery()* a *fillLocation()*

9.6 wheelEvent() a keyPressEvent()

Funkce *wheelEvent()* přibližuje nebo oddaluje mapu v závislosti na pohybu kolečka myši.

keyPressEvent() je funkce, která čeká, až aplikace zaznamená stisk klávesy. Poté podle nastavených pravidel provede určitou akci.

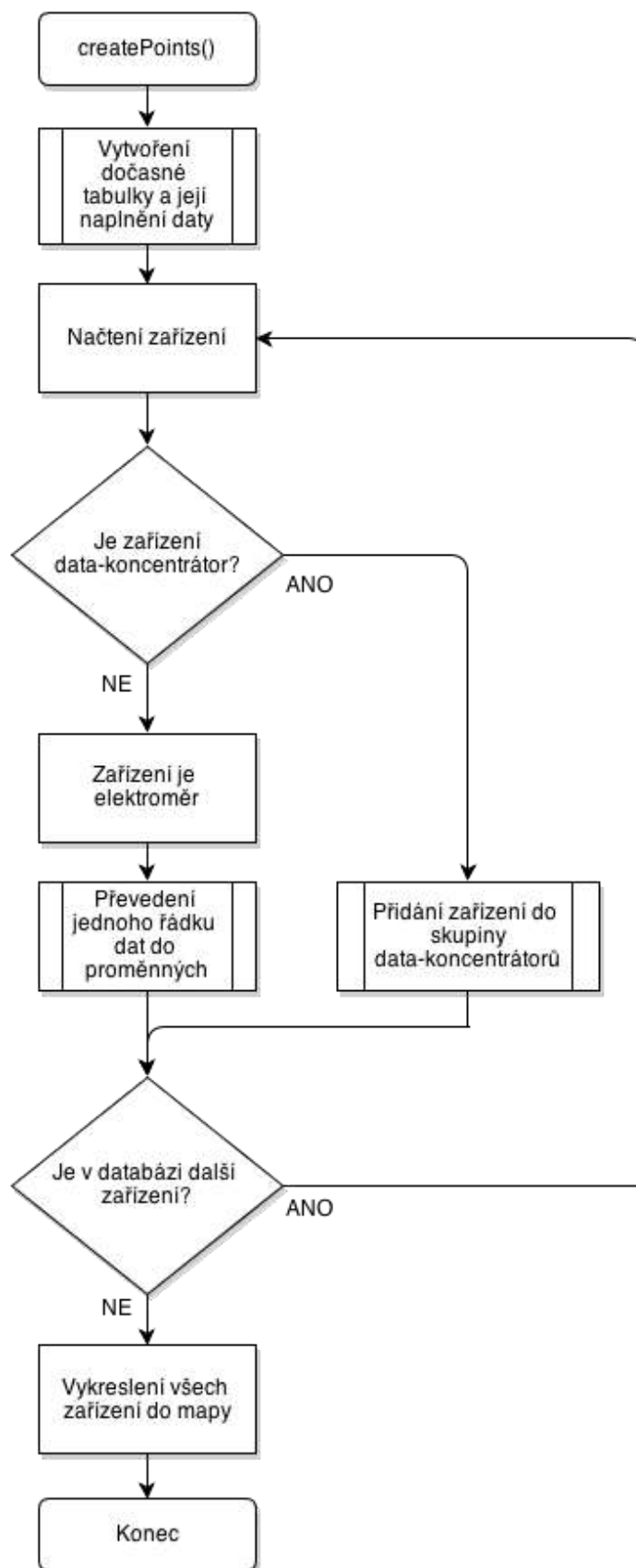
Příklad:

```
if (evnt->key() == 65) // a
{
    mc->scrollLeft();
}
```

Po stisknutí klávesy „a“ se mapa přesune o kousek vlevo.

9.7 createPoints()

createPoints() je funkce potřebná k vykreslení zařízení do mapy chyba (diagram popisující tuto funkci viz Obrázek 9-3). V prvním kroku se vytvoří v databázi dočasná tabulka, která se poté naplní jen potřebnými daty splňující zadané podmínky (např. čas instalace zařízení). Následně se toto zařízení vykreslí do mapy barvou, v závislosti na tom, jestli je zařízení elektroměr (zeleně) nebo data-koncentrátor (černě).



Obrázek 9-3: Vývojový diagram funkce `createPoints()`

10. ZÁVĚR

Cílem bakalářské práce bylo vytvoření aplikace, vyobrazující na mapě umístění elektroměrů a data-koncentrátorů projektu Smart Grid a spojení mezi nimi. V textu je popsáno prostředí software, jeho ovládání a nejdůležitější vnitřní funkce pro správnou činnost programu. Základem tvorby aplikace bylo seznámení se s vývojovým prostředím knihovny Qt pro samotný vývoj a nalezení vhodných doplňkových knihoven pro zprovoznění požadovaných funkcí aplikace.

Pro zobrazení map byl zvolen widget QMapControl, který dokáže získat mapové podklady od poskytovatelů Google, OpenStreetMaps a Yahoo. Hlavní výhodou widgetu je vykreslování bodů na určenou zeměpisnou polohu s různými možnostmi a vlastnostmi. K bodu lze připojit ikona a využít na zakreslení elektroměru či data-koncentrátoru. Při spojení dvou a více bodů do skupiny se mezi nimi vytvoří cesta, kterou je možno různě stylovat.

Veškeré informace a naměřená data zařízení se ukládají do jednoho souboru SQLite databáze, který aplikace aktivně využívá již od jejího spuštění. V průběhu dalšího vývoje aplikace se bude databáze využívat pouze jako dílčí část pro mezivýpočty a veškerá data budou dále zpracovávána serverem.

Aplikace obsahuje funkci pro doplnění chybějících geolokačních údajů. K jejímu zprovoznění bylo zapotřebí seznámení se s projektem Google Maps Geocoding API, který podle zadaných parametrů zašle zpět datové pole ve formátu JSON obsahující veškeré informace o daném místě. Tato funkce je využita i pro vyhledávání libovolné adresy na mapě. Další důležitá funkce má za úkol z načtených datových souborů vypočítat statistická data (hodinové průměry, denní průměry a aktuální hierarchii mezi zařízeními).

V dalších verzích programu bude nutné rozvíjet řešení nedostatků, jako jsou například alternativy přenosu dat z míst, kde není připojení k internetu. Použití off-line map se tak stává jedním z hlavních bodů vývoje. V případě velkých lokalit bude odebíráno velké množství dat, pro jejichž přehlednost by bylo vhodné mít možnost zobrazit jen aktuálně potřebná data místo celé databáze.

11. LITERATURA

- [1] *FUTUR/E/MOTION – Smart Grids* [online]. Dostupné z URL: <http://www.futuremotion.cz/smartgrids/>
- [2] *Smart meter* [online.]. Dostupné z URL: http://en.wikipedia.org/wiki/Smart_meter
- [3] *Qt (knihovna)* [online]. Dostupné z URL: [http://cs.wikipedia.org/wiki/Qt_\(knihovna\)](http://cs.wikipedia.org/wiki/Qt_(knihovna))
- [4] *Qt Reference Documentation* [online]. Dostupné z URL: <http://doc.qt.digia.com/qt/>
- [5] *QMapControl* [online]. Dostupné z URL: <http://www.medieninf.de/qmapcontrol/>
- [6] *QMapControl API* [online]. Dostupné z URL: <http://www.medieninf.de/qmapcontrol/doxygen/index.html>
- [7] *Studijní program Geodézie a kartografie - Kategorie:Qt* [online]. Dostupné z URL: <http://gama.fsv.cvut.cz/gwiki/Kategorie:Qt>
- [8] *C++ Qt Programming (video)* [online]. Dostupné z URL: <http://www.youtube.com/playlist?list=PL2D1942A4688E9D63>
- [9] FRANEK, Lešek. *Data koncentrátor pro chytré sítě: diplomová práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2012. 114 s. Vedoucí práce Ing. Pavel Kučera, Ph.D.
- [10] *Marble* [online]. Dostupné z URL: <http://marble.kde.org/>
- [11] *MapView Class Reference* [online]. Dostupné z URL: http://files.mapeditor.org/doxygen/classTiled_1_1Internal_1_1MapView.html
- [12] *Intelligentní sítě* [online]. Dostupné z URL: http://cs.wikipedia.org/wiki/Intelligentn%C3%AD_s%C3%ADt%C4%9B
- [13] *JavaScript Object Notation* [online]. Dostupné z URL: http://cs.wikipedia.org/wiki/JavaScript_Object_Notation
- [14] *Úvod do JSON* [online]. Dostupné z URL: <http://www.json.org/json-cz.html>
- [15] *The Google Geocoding API* [online]. Dostupné z URL: <https://developers.google.com/maps/documentation/geocoding/>
- [16] *ModemTec* [online]. Dostupné z URL: <http://www.modemtec.cz/>

C. Seznam zkratek

PLC	Power line communication	Přenos dat po elektrické síti
GUI	Graphical User Interface	Grafické uživatelské rozhraní
XML	Extensible Markup Language	Rozšiřitelný značkovací jazyk
API	Application Programming Interface	Rozhraní pro programování aplikací
SQL	Structured Query Language	Strukturovaný dotazovací jazyk
ČEZ	České Energetické Závody	
PRE	Pražská energetika	
CC-BY-SA	Creative Commons Attribution-ShareAlike 2.0	
LAN	Local area network	Lokální síť
WIFI	Wireless LAN	Bezdrátová lokální síť
TCP	Transmission Control Protocol	Transportní protokol
IP	Internet Protocol	Internetový protokol
GPRS	General Packet Radio Service	
RS-485	Recommended Standard 485	Sériová průmyslová sběrnice 485
M-BUS	Meter Bus	Sběrnice M-BUS
JSON	JavaScript Object Notation	JavaScriptový objektový zápis

D. Seznam příloh

Příloha 1. CD s elektronickou verzí práce a zdrojovými kódy aplikace